

Workflow-Management mit Mobile Computing: Ein Überblick

M. Decker, B. Keuter, S. Klink, A. Oberweis, P. Stürzel

Institut AIFB, Karlsruher Institut für Technologie (KIT)
{M.Decker, Keuter, Stefan.Klink, Oberweis, Stuerzel}@kit.edu

Zusammenfassung

Schon seit über drei Jahrzehnten werden neue Workflow-Managementsysteme entwickelt und an aktuelle Standards und technische Möglichkeiten angepasst. Durch die wachsende Flexibilität und Mobilität der Mitarbeiter rücken mobile Geschäftsprozesse weiter in den Vordergrund und werden durch die rasante technische Entwicklung mobiler Endgeräte immer besser unterstützt. Dieser Artikel beschreibt die Grundlagen mobiler Workflow-Managementsysteme (mWfMS) und gibt einen detaillierten Überblick über Systeme. Ausgehend von klassischen Systemen der 1. Generation und modernen Systemen der 2. Generation werden mobil-spezifische Merkmale von mWfMS abgeleitet, und es wird eine Prognose auf zukünftige Systeme der 3. Generation gegeben.

***Schlüsselwörter:** Workflow-Managementsysteme (WfMS), Mobile Geschäftsprozesse, Mobile Computing.*

1 Einleitung

Die zunehmende Verbreitung und der stetige technische Fortschritt von mobilen Geräten führen zur Unterstützung von einzelnen Tätigkeiten fernab von den üblichen statischen Arbeitsplätzen. Jedoch existiert trotz der technisch inzwischen ausgereiften Geräte keine etablierte Integration von mobilen Aktivitäten. Durch diese Integration kann ein Verständnis für den eigentlich benutzten Prozess sowie Ablaufverbesserungen erreicht werden. Workflow-Managementsysteme (WfMS) werden allgemein für die unternehmensinterne ablaufbezogene Integration von Aktivitäten eingesetzt. Geschäftsprozesse beschreiben solche Aktivitäten und verfolgen ein geschäftliches Ziel. Beispielfhaft für klassische Geschäftsprozesse sind Genehmigungsabläufe über mehrere Instanzen hinweg, wobei meist viel Formulararbeit durchgeführt wird. Die Darstellung von Geschäftsprozessen in einem Modell, welches das WfMS zur Ausführung verwendet, wird Workflow genannt. Diese Workflows können (teil)automatisiert ablaufen. WfMS umfassen Komponenten für die Definition, Verwaltung und Ausführung von Workflows.

Die Unterstützung von mobilen Aktivitäten durch ein WfMS steht seit mehreren Jahren im Fokus von verschiedenen Forschungsbemühungen. Systeme dieser Art werden im Folgenden als mobile WfMS bezeichnet (siehe Kapitel 2). Je nach Untersuchungsschwerpunkt finden sich in der Literatur grundlegend unterschiedliche Vorgehensweisen zur Umsetzung von mWfMS. Dies begründet sich nicht zuletzt darin, dass hier zwei eigen-

ständige Forschungsgebiete (Mobile Computing und Workflow Management) in einem Forschungsthema vereint werden.

Der vorliegende Beitrag leitet zunächst mit den Grundlagen in die Thematik der mobilen Workflow-Managementsysteme in Kapitel 2 ein. Kapitel 0 und 0 beschreiben zwei unterschiedliche Generationen von mWfMS. Da aufgrund der Zeitspanne vom Auftreten der mobilen Geräte und Workflow-Technologien bis zum heutigen Stand der Technik bereits viele Veränderungen zu beobachten sind, werden nicht nur aktuelle, sondern auch frühere Entwicklungen vorgestellt. Kapitel 0 schließt mit einer Prognose über zukünftige mobile WfMS ab.

2 Mobiles Workflow-Management

Seit 1993 wird in der Workflow Management Coalition (WfMC) die Standardisierung von Workflows und Business Process Management (BPM) vorangetrieben. Um gemeinsame Komponenten verschiedener Workflow-Systeme mit gleicher Terminologie zu beschreiben, entwickelte die WfMC ein Referenzmodell für Workflow-Managementsysteme (siehe Abb. 1). Auf Basis dieser Arbeiten werden in dem vorliegenden Artikel Workflow-Managementsysteme (WfMS) verstanden. WfMS sind Systeme, die vollständig die Definition, Verwaltung und Ausführung von Workflows durch Software übernehmen und durch die die Ausführungsreihenfolge mit einer Computer-Repräsentation der Workflow-Logik bestimmt wird. Dementsprechend werden *mobile Workflow-Management-Systeme* (mWfMS) von dieser Definition abgeleitet, indem in mWfMS mindestens eine der im Referenzmodell beschriebenen Komponenten auf einem mobilen Gerät vollständig implementiert ist.

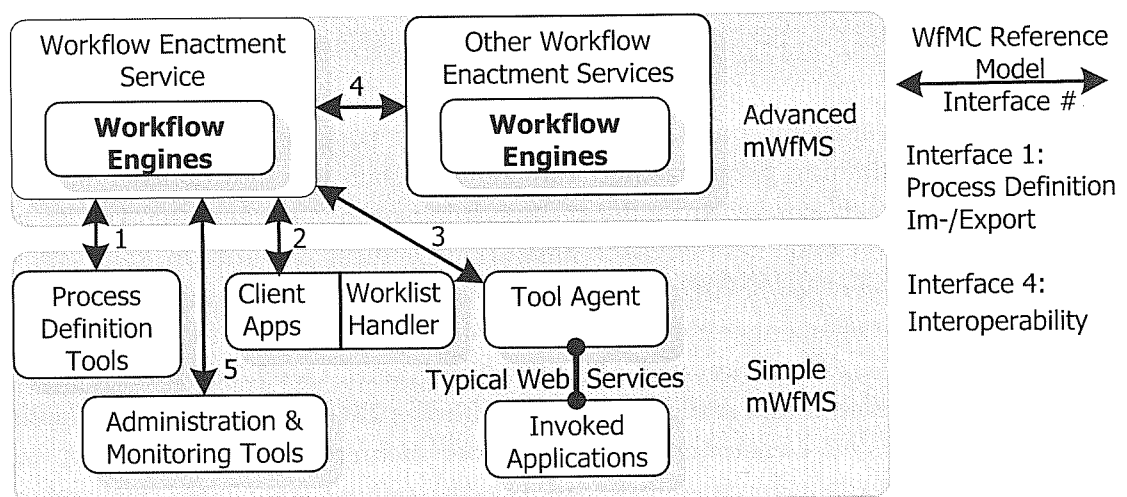


Abb. 1: mobile Workflow-Management-Systeme (mWfMS)

Da die Komponenten verschiedene Aufgaben erfüllen, ist es relevant, die Unterscheidung der jeweiligen mobilen Komponenten zu berücksichtigen. Wird keine Workflow-Engine auf einem mobilen Endgerät ausgeführt, so stellt dies ein *einfaches* mWfMS (mobiler Client) dar. Diese einfachen mWfMS sind vielfältig in ihren Architekturen. Häufig ist nur die GUI als Anwendung für den mobilen Einsatz implementiert. *Erweiterte* mWfMS bieten mindestens in der Komponente der Workflow-Engine eine Unterstützung von mobilen

Geräten und erlauben, Teilprozesse lokal auszuführen. Erweiterte mWfMS stellen höhere Anforderungen an die mobilen Geräte als einfache mWfMS.

Die Entwicklung der mWfMS zeigt, dass mit zunehmender Verbreitung von Serviceorientierten Architekturen (SOA) ein nahezu vollständiger Wandel bzgl. der Schwerpunkte stattgefunden hat. So steht in der *ersten Generation* (1G) von mWfMS in der Regel die Unterstützung mobil-spezifischer Anforderungen der Benutzer im Fokus. Die Systeme der *zweiten Generation* (2G) betrachten deutlich stärker die technische Anbindung von maschinenausführbaren Aktivitäten und nutzen verbreitete Standards wie WS-BPEL und SOAP, welche prinzipiell die Ausführung der Mensch-Aktionen als Spezialfall von Computer-Aktionen beschreiben. Hierdurch erfahren 2G-mWfMS jedoch eine schwächere Unterstützung der Mensch-Aktionen als mWfMS der ersten Generation.

3 Klassische mWfMS (1. Generation)

Die hier der ersten Generation von mWfMS zugeordneten *klassischen* Systeme werden in Publikationen aus den Jahren 1995 bis ca. 2001 beschrieben. Auch wenn einige dieser Ansätze aus heutigem Blickwinkel antiquiert wirken, so werden in diesen Artikeln doch grundlegende mobil-spezifische Aspekte von mWfMS detailliert herausgearbeitet.

Exotica/FMDC ist ein gutes Beispiel für ein WfMS, bei dem ein konventionelles WfMS (IBM Flowmark) um mobile Clients für die Akteure erweitert wurde (Alonso, Günthör, Kamath, Agrawal, Abbadi, & Mohan, 1996). Es wurde bewusst vermieden, Änderungen an anderen Komponenten des Systems vorzunehmen. Die Erweiterung fokussiert sich auf die Unterstützung geplanter Verbindungsabbrüche der mobilen Clients. Hierzu wurde ein entsprechendes Protokoll beschrieben, welches es ermöglicht, dass Akteure Aktivitäten sperren und herunterladen können, bevor sie die drahtlose Verbindung abbauen. Ungeplante Verbindungsabbrüche (Funklöcher) werden jedoch nicht berücksichtigt.

Jing et al. beschreiben das WHAM-System (Jing, Huff, Hurwitz, Sinha, Robinson, & Feblowitz, 2000). Als mobil-spezifische Erweiterungen sieht dieses System insbesondere eine Worklist-Darstellung vor, bei der die zu erledigenden Aktivitäten in einer geographischen Karte eingetragen werden. Ein mobiler Akteur kann sich bei seiner Wahl für die nächste zu bearbeitende Aktivität danach richten, in welcher Entfernung die entsprechenden Orte liegen. Weiter unterscheidet WHAM zwei aufeinander aufbauende Formen der Zuordnung der anstehenden Aktivitäten zu den Akteuren: das *global resource assignment* wird auf der zentralen Server-Infrastruktur durchgeführt und berücksichtigt auch den Aufenthaltsort des mobilen Akteurs, wohingegen das *local resource assignment* auf dem mobilen Endgerät des Akteurs durchgeführt wird: es priorisiert die in der Worklist aufgeführten Themen nach Informationen, die nur dem Endgerät vorliegen, z.B. verfügbare Ausrüstungsgegenstände, Verbrauchsmaterialien wie Ersatzteile oder persönlichen Präferenzen des Akteurs. Für die Ressourcen-Zuordnung bietet WHAM u.a. die sog. *optimistische Zuordnung* an, bei der einem mobilen Akteur auch dann eine Aktivität zugeordnet werden kann, wenn er gerade nicht mit dem System verbunden ist. D.h. in diesem Fall ist es nicht bekannt, ob er sich tatsächlich in der Nähe des Ortes befindet, wo die entsprechende Aufgabe abzuarbeiten ist. Die optimistische Zuordnung wird aber nur für als *steady* eingeordnete Akteure angewendet, also für Akteure, die sich innerhalb einer bestimmten Zeitspanne nicht über ein bestimmtes Gebiet hinaus bewegt haben. Ist dies nicht der Fall, wird der Akteur als *unsteady* eingeordnet, so dass er nur dann Aktivitäten

zugeordnet bekommt, wenn er gerade mit dem System verbunden ist, also sein tatsächlicher Aufenthaltsort auch bekannt ist. Weiter beinhaltet WHAM die Funktionalität, auf das mobile Endgerät zu ladende Workflow-Daten in der Größe reduzieren zu können, z.B. durch Kompressionsverfahren oder Verringerung der Auflösung; dieses Feature zielt aber vorrangig auf Bitmap-Daten wie eingescannte Dokumente oder Fotografien ab.

Der Ansatz, die einzelnen Aktivitäten in der Arbeitsliste eines Akteurs in einer geographischen Karte darzustellen, wird von Leoni et al. in (Leoni, der, & Hofstede, 2008) und von Brown & Paik in (Brown & Paik, 2005) aufgegriffen; die geographische Karte ist in beiden Arbeiten allerdings nur ein Spezialfall der Visualisierung von Arbeitslisten. Es können auch abstrakte Distanzmaße verwendet werden, z.B. um die benötigte Expertise eines Akteurs für eine bestimmte Aufgabe darzustellen.

Domingos et al. (Domingos, Martins, Prequica, & Duarte, 1999) stellen ein mWfMS vor, bei dem die mobilen Endgeräte über Messaging-Kanäle mit der stationären Infrastruktur verbunden sind. Messaging-Kanäle bieten asynchrone Kommunikation, welche sich besser als das herkömmliche *Request-Response*-Muster für mobile Szenarien eignen, da das Gesamtsystem dann unempfindlicher im Bezug auf unvorhergesehene Kommunikationsabbrüche (Funklöcher) wird. Das System unterstützt zwei grundlegende Arten der Zuordnung von Aktivitäten zu Akteuren, namentlich das *geschlossene Verfahren* und das *offene Verfahren*. Das *geschlossene Verfahren* ist für den Fall gedacht, dass eine zuverlässige Kommunikationsverbindung zwischen dem mobilen Client und dem Back-End besteht; dies ist etwa dann gegeben, wenn der Akteur sich während der morgendlichen Einsatzbesprechung im Unternehmen befindet. Der *Planer* kann dann einzelne Aufgaben an mobile Akteure zuweisen. Die damit verbundene Sperre verfällt aber, wenn der mobile Client nicht in bestimmten Zeitabständen ein *Keep-Alive*-Signal übermittelt. Das *offene Verfahren* wird benutzt, um eine neue Aktivität (z.B. ein dringender Reparatursatz) einem Akteur zuzuordnen, während dieser sich schon im Einsatz befindet. Für diesen Fall steht ein spezieller Messaging-Kanal bereit, über den der Planer und die Akteure die Zuordnung der neuen Aktivität aushandeln können.

Stormer & Knorr (Stormer & Knorr, 2001) beschreiben mit *AWA/PDA* ein mWfMS, das sich der Agenten-Technologie bedient, um den besonderen Anforderungen mobiler Workflow-Szenarien gerecht zu werden. Agenten werden hier als Software-Module verstanden, die bis zu einem gewissen Grad autonom zur Erreichung eines bestimmten Zieles handeln können und hierfür untereinander Nachrichten austauschen. Bei mobilen Agenten liegt insbesondere Code-Mobilität vor, d.h. eine Agenten-Instanz kann während ihrer Ausführung auf verschiedenen physikalischen Rechnern verschoben werden. *AWA/PDA* basiert auf mehreren Typen solcher mobilen Agenten: eine laufende Workflow-Instanz wird durch einen Workflow-Agenten repräsentiert; für die Beschreibung des zugehörigen Workflow-Schemas ist der Prozess-Agent zuständig. Für jede durch einen Akteur abzuarbeitende Aktivität innerhalb eines Workflows gibt es einen Task-Agenten, der alle hierfür benötigten Daten (z.B. Formulare, Dokumente) kapselt. Die Zuordnung von Aktivitäten an einzelne Akteure wird durch den Worklist-Agenten vorgenommen. Der Task-Agent kann auf ein mobiles Endgerät migrieren, so dass die jeweilige Aktivität auf dem Endgerät autark ausgeführt werden kann. Auf dem Endgerät selbst läuft der Personal-Agent, der allerdings kein mobiler Agent ist; seine Aufgabe ist die Kommunikation des Nutzers mit dem Task-Agenten, also etwa die Abfrage von Formulareingaben über die graphische Benutzeroberfläche.

4 Moderne mWfMS (2. Generation)

Den modernen mWfMS der 2. Generation werden Systeme aus Publikationen ab ca. 2002 zugeschrieben. Diese zeichnen sich durch einen speziellen Entwurf bzw. einer auf Standards aufbauenden Entwicklung aus. Weitere Charakteristika sind die XML-basierte Implementierung und die Verfolgung des SOA-Paradigmas.

Sliver ist ein repräsentatives Beispiel für ein mWfMS der 2. Generation (Hackmann, Haitjema, Gill, & Roman, 2006). Die Autoren haben sich zum Ziel gesetzt, ein Groupware-System auf Basis von BPEL (*Business Process Execution Language*) zu entwickeln. Die herkömmlichen Implementierungen einer BPEL-Engine beruhen auf einem Java-Applikationsserver (Tomcat, JBoss, etc.) und sind für den Einsatz auf den meisten mobilen Endgeräten nicht geeignet. Viele mobile Betriebssysteme unterstützen Java ME (*Mobile and Embedded Java Platform*); allerdings unterscheidet sich diese grundsätzlich von Java SE (*Java Standard Edition*). Daher ist es nicht möglich, Java SE Programme auf einer Java ME Plattform auszuführen. Java SE ist zu umfangreich für den Einsatz auf mobilen Endgeräten. Die Autoren haben demzufolge eine neue Architektur entworfen und umgesetzt. Eine Transportschicht tauscht Nachrichtenobjekte in Form von XML aus. Die Daten werden mit Hilfe der XML-Parser- und der SOAP-Schicht in Java-Objekte konvertiert. Der BPEL-Server verwendet diese Komponenten. Zum Beispiel benutzt der BPEL-Server XML-Parser, um BPEL in ausführbare Prozesse zu übersetzen. *Sliver* besitzt insgesamt eine Codegröße von 114KB und unterstützt 16 der 20 wichtigsten Workflow-Patterns (W.M.P. v. d. Aalst, 2003). Die Autoren zeigen, dass mit von Hand geschriebenen XML-Parsern (ohne Validierungsfunktionen) die Implementierung einer mobilen BPEL-Engine möglich ist.

Chen et al. stellen mit dem System *FollowMe* eine Workflow-Engine mit Unterstützung von *pvPDL* (*Pervasive Process Definition Language*) vor (Chen, Bu, Li, Tao, & Lu, 2006), welche eine von den Autoren speziell für kontextsensitive Applikationen entworfene Beschreibungssprache ist. Die entwickelte Engine basiert auf dem OSGi-Rahmenwerk, das wiederum einen Applikationsserver (*Classloader*) voraussetzt. Der Nachweis, dass ein Applikationsserver auf einem mobilen Gerät lauffähig ist, wird jedoch nicht erbracht. Das Kontextmodell wird mit Hilfe von Ontologien beschrieben. Der Hauptfokus ihrer Arbeit ist einerseits die Unterstützung von kontextsensitiven Applikationen und andererseits die Beschleunigung der Softwareapplikationsentwicklung auf Basis von Workflows. Dies wird in ihrer Evaluation gezeigt, indem eine Applikation mit und eine ohne Rahmenwerk implementiert wird. Ergebnis ist, dass bei der Workflow-Lösung der personelle und zeitliche Aufwand niedriger ist. Als Szenario wird ein *Meeting Assistant* vorgestellt, in dem das System einen Dozenten während eines Vortrags unterstützt.

Schmidt et al. präsentieren in (Schmidt, Kapitza, & Hauck, 2007) den Entwurf eines Systems, das mobile Prozesse im *Ubiquitous Computing* unterstützen soll. Dabei liegt der Fokus der Autoren auf kontextsensitiven Prozessen. Dieses *Middleware*-System soll es ermöglichen, zur Laufzeit Webservices inklusive der Implementierung verschiedener Zustände und Schnittstellen über sogenannte *Facetten* auszuführen. Orchestriert werden die Prozesse mit Hilfe von BPEL. Aus dem Artikel wird die genaue Funktionsweise allerdings nicht ersichtlich. Es wird kein Prototyp vorgestellt. Als Anwendungsszenario wird ein mobiler Journalist beschrieben, der laut Autoren nur die Applikationslogik für jede *Facette* erstellen muss, während der Anwendungscode auf Basis des mobilen Prozesses

automatisch (auch mit Mensch-Interaktionen) erstellt wird. Im Ausblick ihrer Arbeit wird eine Implementierung angekündigt.

Ein mobiles Ad-hoc-Netzwerk (MANET) ermöglicht es, zur Laufzeit mobile Endgeräte zum Netzwerk hinzuzufügen, entsprechendes Routing zu implementieren und eine Kommunikation ohne spezielle Infrastruktur zur Verfügung zu stellen. Sen et al. stellen in (Sen, Hackmann, Haitjema, Roman, & Gill, 2007) ein WfMS auf Basis eines MANETs vor. Ein Baustellenszenario mit Vorarbeitern, die z.B. die Arbeitszuteilung vornehmen, beschreibt detailliert, wie solch ein WfMS betrieben werden kann. Die Autoren berechnen beispielsweise die Einteilung der Aufgaben nach der Morgenbesprechung mit Hilfe von Heuristiken, die verschiedene Ressourcen berücksichtigen. Das System ist auf Basis von CAST (Roman, Handorean, & Sen, 2006) implementiert worden, um die Koordination zwischen den verschiedenen Teilnehmern bzw. Endgeräten bezüglich Einsatzort und -zeit in den Workflows zu verwalten. Mit fünf Netzwerktransaktionen haben die Autoren gezeigt, dass ein vollständiges Ad-Hoc-WfMS möglich ist. Ergebnisse der Evaluation zeigen eine lange Laufzeit der heuristischen Algorithmen. Allerdings werden zur Beschleunigung auch Lösungsansätze, wie z. B. die Begrenzungen des Entscheidungsraums, präsentiert.

Kunze et al. stellen für mobile Prozesse eine *Execution Engine* vor, welche auf der DEMAC-Middleware aufbaut (Kunze, Zaplata, & Lamersdorf, 2007). Als Szenario für mobile Geschäftsprozesse beschreiben die Autoren eine Auto-Panne, bei deren Behebung ein mobiles Informationssystem die Pannenhelfer in Kooperation mit der Pannen-Meldezentrale unterstützt.

Gunarathne et al. beschreiben eine eigens implementierte BPEL-Engine, welche einbettbar, skalierbar und erweiterbar ist (Gunarathne, Premala, Wijethilake, Kumara, & Kumar, 2007). Die meisten BPEL-Engines sind laut Autoren schwerfällig, komplex und nicht erweiterbar. Kontextinformationen werden strukturiert im *Information Model* abgelegt, wobei das *Process Model* dazu verwendet wird, um die eigentlichen Workflows zu speichern. Ein *Kernel* hilft dabei, dass diese Engine auch auf mehreren CPUs ausgeführt werden kann. Die Implementierung erfolgt in Java-Threads. Allerdings zeigen die Autoren nicht, ob diese Engine, die auf Axis2 aufbaut, auch mit mobiler Hardware ausgeführt werden kann. Axis2 basiert auf Java 1.5, wobei auch eine reine C-Implementierung existiert, mit der eine Portierung auf ein mobiles Endgerät sicherlich möglich wäre.

Pajunen und Chande beschreiben in (Pajunen & Chande, 2007) ein vollständiges mWfMS, das eine Plattform zum Einspielen und Verwalten von Prozessen auf Basis von Diensten zur Verfügung stellt. Gerade die Unterstützung von Standards wird in diesem Paper zum Hauptfokus und es wird gezeigt, welche Vorteile daraus entstehen. Zunächst werden die Interaktion und die Integration einer Workflow-Engine als Hauptindikator zur Unterstützung von mobilen Prozessen identifiziert und folgerichtig auch implementiert. Dabei ist diese Workflow-Engine an die Funktionen und Kapazitäten eines *Nokia S60* angepasst worden. Mit einem Szenario werden Anforderungen an das System formuliert mit dem Ergebnis, dass mehr Forschungsarbeit in die Integration lokaler Applikationen in diese Workflow-Engine fließen soll. Außerdem sollen die unterstützten Standards auf ihre Zukunftsfähigkeit hin untersucht werden. Einer davon ist die Unterbrechungsunterstützung. Sie wird als eines der Hauptkriterien angesehen, das zwar schon frühzeitig in WfMS charakterisiert wurde (Alonso, Günthör, Kamath, Agrawal, Abbadi, & Mohan, 1996), aber bis heute noch nicht vollständig in eine Prozesssprache bzw. ein

WfMS integriert ist. Mit der Forderung einer von herkömmlichen Systemen unterschiedlichen Zugriffskontrolle wird der Ausblick dieses Artikels abgeschlossen.

Mnaouer et al. stellen in (Mnaouer, Shekhar, & Liang, 2004) die Entwicklung einer Workflow-Engine unter die Prämisse der schnellen Applikationsentwicklung. Dabei wollen die Autoren auch Standards unterstützen und mit ihrem Ansatz das „Gestänge des Modells“ bilden, um eine interoperable, flexible und einfach zu implementierende mobile Entwicklungsumgebung als Open Source zu veröffentlichen. Die unterstützten Standards sind wie folgt: WSFL (Vorgänger von BPEL), XML, SOAP, UDDI. Als Motivation ihres Ansatzes wird ein zukünftiges mobiles System mit einer dynamischen Workflow-Engine und vielen sogenannten *Core Services* im *Backend* vorgeschlagen. Die Implementierung der Wf-Engine bzw. die Unterstützung der genannten Standards wird mit Hilfe von kXML (kleiner XML Parser) und kSOAP (SOAP API für Java ME) in Java ME implementiert. Die wichtigsten Designkriterien der Autoren sind Sicherheit, Verbindungsmodell, einfache Entwicklung, Interoperabilität und Service-Orientierung.

5 Zukünftige mWfMS (3. Generation)

Im Folgenden fassen wir spezifische Merkmale von mWfMS der letzten Kapitel zusammen:

- Behandlung von geplanten und ungeplanten Verbindungsabbrüchen.
- Berücksichtigung von Kontextparametern bei Aufgabenzuordnungen.
- Reduzierung des zu übertragenden Datenvolumens.
- Standardisierte Benutzungsschnittstellen.
- Integration (Verknüpfung) von Anwendungsprogrammen auf mobilen Endgeräten.

mWfMS der dritten Generation basieren auf einer festen Infrastruktur wie z. B. dem heutigen Mobilfunknetz. Mit Hilfe von IPv6 werden diese Systeme in viele dynamische Gruppen untergliedert. Diese Systeme werden als „nomadische“ Systeme bezeichnet, weil sie eine statische Infrastruktur einsetzen, jedoch zugleich eine dynamische, z.B. von Orts-Constraints abhängige Zugehörigkeit besitzen. In diesen Systemen gibt es weiterhin eine zentrale Kommunikation (z. B. über den Workflow-Manager); allerdings sind sie von außen betrachtet dezentral organisiert. Dies kann z.B. bedeuten, dass im Falle des Ausfalls der zentralen Einheit andere Systemkomponenten zur zentralen Einheit werden. Einhergehend mit nomadischen Systemen oder Netzwerken ist somit eine automatische bzw. ad-hoc-basierte Anbindung von neuen Clients bzw. Ressourcen zur Laufzeit eingeschlossen. Dienstinformationen, z. B. bezüglich des Wetters, sind fast immer ortsabhängig zu beziehen.

Ein weiterer wichtiger Punkt für zukünftige mobile Systeme ist deren Steuerung. Das Schreiben von Text mit einem Mobilfunktelefon ist mühsam. Aktuell haben hier Mobilgerätehersteller gezeigt, dass eine Steuerung per Multitouch eine erhebliche Verbesserung der Bedienung solcher Geräte darstellen kann. Weitere Steuerungsformen, wie z. B. über Gesten oder Sprache bzw. die ad-hoc Anbindung von User Interfaces, die stationär zur Verfügung stehen, werden mWfMS der Zukunft auszeichnen.

Eine automatische Anbindung von Ressourcen, also weiteren Sensoren oder anderen Geräten, ist heutzutage noch nicht durchgängig realisiert. Obwohl schon seit Jahrzehnten der Begriff *Plug-and-Play* verschiedene computeraffine *Communities* durchwandert, ist *Plug-and-Play* im Bereich Workflow-Management noch nicht umgesetzt. *Bonjour* und

sogenannte *Zero-Configuration*-Protokolle ermöglichen es, beliebige IP-basierte Geräte, wie z. B. Drucker, automatisch anzubinden. Diese Protokolle unterstützen jedoch keine Workflows.

Der größte Mehrwert von 3G-Systemen wird jedoch durch die Darstellung von Kontextinformationen erreicht. Heutzutage sind Anfragen möglich, die einen Prozess inkl. Kontextinformationen während seiner Laufzeit auswerten. *Complex Event Processing* (CEP) hat in der Geschäftsprozessausführung eine neuartige Generierung von Informationen ermöglicht (Anicic, Fodor, Stühmer, & Stojanovic, 2009). Diese Technik ist in einzelnen Systemen implementiert und damit für viele Benutzer sehr unhandlich (Gardner, 2009). Allerdings werden sich in Zukunft Anfragesprachen und Benutzungskonzepte weiterentwickeln, so dass auch auf Mobilgeräten komplexere Ereignisse definiert und ausgewertet werden können. Gerade die Kombination von verschiedenen Kontexten zu *höheren* Kontexten ermöglicht eine Fülle von neuen Bedingungen, die im System wiederum zur Verbesserung benutzt werden kann. Der Aspekt der Kontextgenerierung wird aktuell auch in anderen Forschungsgebieten, z.B. Data Mining und Information Retrieval, untersucht (Haghighi, Zaslavsky, Krishnaswamy, Gaber, & Loke, 2009). Allerdings ist eine Adaption in Richtung der WfMS bisher noch nicht erfolgt.

Bei 2G-Systemen haben wir in Abschnitt 0 festgestellt, dass sich die Autoren an Standards orientieren, aber dabei einen zentralen Aspekt eines WfMS vernachlässigen – nämlich den Menschen selbst. Systeme der Zukunft werden sich auf den Menschen besinnen. Sie sollen einen Mehrwert für den Menschen darstellen und nicht Zeit durch unhandliche Bedienung vergeuden. Die eigentlichen menschlichen Arbeitsschritte in einem Ablauf können nicht vollständig als automatisierbare Aktivität betrachtet werden. Gerade *Personal Workflows* zeigen, dass es hier ein starkes Forschungsinteresse gibt (Hwang & Chen, 2003). Eine allgegenwärtige Integration eines (mit den Geräten einhergehenden) *Personal Workflows* in eine bestehende dynamische Gruppe, stellt eine essentielle Säule der Prognose zur 3. Generation von mWfMS dar.

6 Literaturverzeichnis

- [1] C. Bussler, "User Mobility in Workflow-Management-Systems", *Proceedings of the Telecommunications Information Networking Conference (TINA '95)*, Melbourne, Australia, 1995
- [2] G. Alonso, et al., "Exotica/FMDC: A Workflow Management System for Mobile and Disconnected Clients", *Distributed and Parallel Databases*, vol. 4, 229-247, 1996
- [3] J. Jing, et al., "WHAM: Supporting Mobile Workflows and Applications in Workflow Environments", *Proceedings of 10th International Workshop on Research Issues in Data Engineering (RIDE, 31-38)*, 2000
- [4] M. d. Leoni, W. M. P. v. der Aalst, and A. H. M. t. Hofstede, "Visual Support for Work Assignment in Process-Aware Information Systems", *Proceedings of Business Process Management 2008*, 67-83, 2008
- [5] R. Brown and H.-Y. Paik, "Resource-Centric Worklist Visualisation", in *Proceedings of*

- [6] H. J. Domingos et al., "A Workflow Architecture to Manage Mobile Collaborative Work", *Encontro Português de Computação Móvel 1999 (EPCM 99)*, 49-61, 1999
- [7] H. Stormer and K. Knorr, "PDA- and Agent-based Execution of Workflow Tasks", in *Proceedings of Informatik 2001*, 968-973, 2001
- [8] G. Hackmann et al., "Sliver: A BPEL Workflow Process Execution Engine for Mobile Devices", *4th Intl. Conf. on Service Oriented Computing*, LNCS vol. 4294, 503-508, 2006
- [9] W. M. P. v. der Aalst et al., "Workflow Patterns," in *Distributed and Parallel Databases*, vol. 14(1), 5-51, 2003
- [10] S. Chen et al., "Toward Context-Awareness: A Workflow Embedded Middleware", *3rd Intl. Conf. on Ubiquitous Intelligence and Computing*, LNCS vol. 4159, pp. 766-775, 2006
- [11] H. Schmidt et al., "Mobile-process-based ubiquitous computing platform: a blueprint", *Proceedings of the 1st Workshop on Middleware-Application Interaction*, 25-30, ACM, 2007
- [12] R. Sen, G. Hackmann, M. Haitjema, G.-C. Roman, and C. D. Gill, "Coordinating Workflow Allocation and Execution in Mobile Environments", *Coordination Models and Languages*, LNCS vol. 4467, 249-267, 2007
- [13] G. C. Roman, R. Handorean, and R. Sen, "Tuple space coordination across space and time", *Coordination Models and Languages*, LNCS vol. 4038, 266-280, 2006
- [14] C. P. Kunze, S. Zaplata, and W. Lamersdorf, "Mobile Processes: Enhancing Cooperation in Distributed Mobile Environments", *Journal of Computers*, vol. 2(1), 1-11, 2007
- [15] T. Gunarathne et al., "BPEL-Mora: Lightweight Embeddable Extensible BPEL Engine", *Emerging Web Services Technology*. Birkhäuser, 2007
- [16] L. Pajunen and S. Chande, "Developing Workflow Engine for Mobile Devices", *Proceedings of the 11th IEEE Intl. Conf. on Enterprise Distributed Object Computing*, p. 279, 2007
- [17] A. B. Mnaouer et al. "A Generic Framework for Rapid Application Development of Mobile Web Services with Dynamic Workflow Management", *Proceedings of the 2004 IEEE International Conference on Services Computing*, 165-171, 2004
- [18] D. Anicic et al., "Event-driven Approach for Logic-based Complex Event Processing", in *Proc. of the 12th IEEE Intl. Conf. on Computational Science and Engineering*, 56-63, 2009
- [19] D. Gardner, "Progress gives CEP a performance boost with multi-core support on

Apama", <http://www.it-director.com/business/change/content.php?cid=11239>, (besucht am 29.09.09)

- [20] P. D. Haghigh et al., "Context-aware adaptive data stream mining", *Intelligent Data Analysis*, vol. 13(3), 423-434, 2009
- [21] S.-Y. Hwang and Y.-F. Chen, "Personal Workflows: Modeling and Management", in *Mobile Data Management*, LNCS vol. 2574, 141-152, 2003